FUNCTIONS 106

Controller 108

110-0
USB Host 102

Second Host 104

110-1

110-2

FUNCTIONS 106

References Cited

U.S. PATENT DOCUMENTS

4,641,261 A 2/1987 Dwyer et al.
4,701,913 A 10/1987 Nelson
4,713,757 A 12/1987 Davidson et al.
4,866,606 A 9/1989 Kopetz
4,890,222 A 12/1989 Kirk
5,388,249 A 2/1995 Hotta et al.
5,392,421 A 2/1995 Lenarttsson
5,402,394 A 3/1995 Turski
5,416,909 A 5/1995 Long et al.
5,428,748 A 6/1995 Davidson et al.

FOREIGN PATENT DOCUMENTS

EP 0890905 A2 1/1999

OTHER PUBLICATIONS

Axelson, Jan; “USB Complete”; Independent Publishers Group; Third Edition; 2005; various pages.*

Primary Examiner — Ryan Stiglic

ABSTRACT

A controller circuit can provide communication paths between multiple host devices and at least one function interface (I/F), where a function I/F can allow access to a predetermined circuit function. The controller circuit can include an endpoint buffer circuit having a plurality of storage locations configurable as endpoints according to a predetermined data transmission protocol and a data switching circuit coupled to the endpoint point buffer circuit. The data switching circuit is configurable to provide communication paths that enable a first host I/F and a second host I/F to access at least a same function I/F, and enable the first and second host I/Fs to communicate with one another.

22 Claims, 11 Drawing Sheets
* cited by examiner
FIG. 6

FIG. 7
FIG. 8

FIG. 9
FIG. 10

FIG. 11
FIG. 12A (BACKGROUND ART)

FIG. 12B (BACKGROUND ART)

FIG. 12C (BACKGROUND ART)
ARCHITECTURES FOR SUPPORTING COMMUNICATION AND ACCESS BETWEEN MULTIPLE HOST DEVICES AND ONE OR MORE COMMON FUNCTIONS

TECHNICAL FIELD

The present invention relates generally to architectures that provide communication paths between host devices and functions, and more particularly to architectures for accommodating communication paths between multiple hosts and one or more shared functions.

BACKGROUND OF THE INVENTION

Communication protocols can allow a "host" computing devices (for example a personal computer, laptop computer, etc.) communicate with one or more "secondary" devices. For example, media files can be transferred between media players and a host, image files can be transferred from a camera to a host, or input devices can provide input data for the host (e.g., mouse, game controller, etc.). One popular communication protocol that is included in the Universal Serial Bus (USB) Specification.

To better understand various aspects of the disclosed embodiments, conventional systems for interconnecting secondary devices with a host will now be described with reference to FIGS. 12A to 12C.

FIG. 12A shows a first conventional USB system, designated by the general reference character 1200. Conventional system 1200 can include a USB host device 1202, a USB controller 1204 and a circuit that executes a predetermined function 1206. USB host device 1202 communicates with USB controller 1204 via a USB connection 1208. USB controller 1204 can access function 1206 in response to requests from USB host device 1202. In such an arrangement, only the USB host device 1202 accesses the given function 1206. That is, the function 1206 is not common to more than one host device.

FIG. 12B shows a second conventional USB system, this one designated by the reference character 1220. Conventional system 1220 includes a USB host device 1222 connected to a USB controller 1224, via a USB connection 1228, in a similar fashion to that of FIG. 12A. However, in FIG. 12B a second host device 1223 is connected to USB controller 1224 via an interface 1225. In such an arrangement, USB 1222 host and second host 1223 can communicate with one another. However, the two host devices do not share a common function.

FIG. 12C shows a third conventional USB system, this one designated by the reference character 1230. Conventional system 1230 includes the same arrangement as FIG. 12B, but includes function 1236. In such an arrangement, second host 1233 can have direct access to function 1236. USB host 1232 can access function 1236, but such access must be in second host 1233. As a result, the second host 1233 typically requires additional software, and if necessary, computing power, as requests to access function 1236 by first host 1232 must first be processed by second host 1233.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a block diagram of a system according to one embodiment of the invention.
FIGS. 2A to 2C show a controller device and associated configurations according to an embodiment of the invention.
FIG. 3 shows a controller device according to another embodiment of the invention.
FIG. 4 shows a controller device according to yet another embodiment of the invention.
FIGS. 5A to 5F show various configurations for the controller device of FIG. 4.
FIG. 6 shows an example of a configurable function interface (I/F) that can be included in the embodiments.
FIG. 7 shows another example of a configurable function I/F that can be included in the embodiments.
FIG. 8 shows an example of a configurable host I/F that can be included in the embodiments.
FIG. 9 shows an example of a configurable clock circuit that can be included in the embodiments.
FIG. 10 shows an example of a configuration circuit that can be included in the embodiments.
FIG. 11 shows an example of a host interface circuit that can be included in the embodiments.
FIGS. 12A to 12C show examples of conventional Universal Serial Bus (USB) systems.

DETAILED DESCRIPTION

Various embodiments will now be described in detail that show methods and devices that can allow for communication paths to be created between two hosts, as well as between the hosts and a common function. Such communication paths can be essentially independently of another. As a result, accesses to a function by either host can be "direct" (i.e., not pass through the other host). Further, such an arrangement can allow hosts to access multiple functions, and/or allow multiple logical communication endpoints to be created from a single physical communication endpoint. Such arrangements may provide significant communication speeds over conventional arrangements like that shown in FIGS. 12A to 12C, without additional software or hardware, as in the case of FIG. 12C.

Referring now to FIG. 1, a system according to a first embodiment is shown in a block diagram and designated by the reference character 100. A system 100 can include a first host device 102, a second host device 104, a functions circuit 106, and a controller circuit 108. A first host device 102 can send commands, data and receive responses and data according to a predetermined protocol. In the very particular example shown, a first host device 102 can be a Universal Serial Bus (USB) type host.

A second host device 104 can also send commands and data, and receive responses and data. While a second host device 104 could be a USB type host, a second host device 104 can also operate according to a different protocol than first host device 102, and/or can have an entirely different interface than first host device 102. As but two very particular examples, if first host device 102 communicates via a serial interface (e.g., USB), second host device 104 can communicate via a parallel interface or different type of serial interface.

Functions circuit 106 can provide one or more predetermined function. However, unlike a conventional case like that of FIG. 12C, one or more functions executable by functions circuit 106 can be accessed by either first host device 102 or second host device 104 directly, and not via the other host device. In very particular embodiments, a functions circuit 106 can provide a storage function and/or an input/output (I/O) function. A storage function can be a function that receives data for storage on a storage media. As but one example, a storage function can store media files, even more particularly, store media files on a solid state nonvolatile media, such as "flash" EEPROM memory. An I/O function...
can allow a host device to send data to, and receive data from, another device or circuit, in order to execute a predetermined operation. As but a few examples, an I/O function can be a modem, input device, graphical display, cell phone, computer, personal digital assistant (PDA), etc.

A controller circuit 108 can provide configurable communication paths between first host device 102, second host device 104 and functions circuit 106. Such communication paths are shown as 110-0, 110-1 and 110-2. Communication path 110-0 can be a first host-to-second host (H1→H2) path formed between first host 102 and second host device 104 that does not access functions circuit 106. Communication path 110-1 can be a second host-to-function (H2→F) path formed between second host 104 and functions circuit 106 that does not access first host device 102. In a similar fashion, communication path 110-2 can be a first host-to-function (H1→F) path formed between first host device 102 and functions circuit 106 that does not access second host device 104.

In this way, a controller circuit 108 can provide direct communication paths between host devices and between the host devices and one or more common functions. Such direct communication paths (e.g., paths not through other devices or functions) can allow for faster and/or more efficient data throughput or function execution. Further, such an arrangement can allow for greater versatility in a secondary device, as multiple functions can be accessed by any of multiple host devices.

Referring now to FIGS. 2A to 2C a controller circuit according to another embodiment is shown in a block schematic diagram, and designated by the general reference character 200. In one very particular arrangement, controller circuit 200 can correspond to that shown as 108 in FIG. 1.

A controller circuit 200 can include an endpoint (EP) storage circuit 202, a switch/control circuit 204, configuration circuit 206, a first host interface (I/F) 208, a second host I/F 210, and a function I/F 212. An EP storage circuit 202 can include storage locations divided, or dividable into multiple EP locations. Each such EP location can store a quantifiable amount of data (e.g., data of a given packet length, or payload size). Such EP locations can also be assigned an identification value (e.g., EP number), and configured for a particular direction. In the example of FIGS. 2A to 2C, EP location directions are made with respect to a host I/F 208. Thus, data received from a direction from first host I/F 208 can be considered an “OUT” direction. Data being sent toward first host I/F 208 can be considered an “IN” direction. Further, and as will be described in more detail below, a size of each EP location can also be configurable.

It is noted that an endpoint can be a destination or source of information commonly identifiable by all entities participating in a communication path (e.g., device, threads, etc.). In the very particular example of the Universal Serial Bus (USB) protocol, an endpoint can be a uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between a host and a function. Each USB endpoint can have an endpoint number as well as a direction of flow.

In very particular arrangements, the physical implementation of an endpoint can include an endpoint buffer. An endpoint buffer can be a memory circuit that stores data arriving from a host (configured in the OUT direction), or that stores data that can be read out to a host (configured in the IN direction). Thus, a physical endpoint can be a circuit that can actually stores data in a communication flow. In particular embodiments, such data can be in a predetermined packet format. According to particular embodiments of the invention, an endpoint buffer can be a first-in-first-out memory (FIFO), with start and end pointers that can dictate where data can be read from or written to. Such endpoint FIFOs can include flags to indicate when the FIFO is ready to be read from or written to.

A physical endpoint is a buffer where data will be stored. Logical endpoints are USB endpoints that can be mapped to any physical endpoint. Physical endpoint can be dedicated to a logical endpoint or shared between multiple logical endpoints. All logical endpoints that map to a single physical endpoint share the buffer(s).

Referring still to FIG. 2, a switch/control circuit 204 can provide data paths between the various I/Fs (208, 210 and 212) and EP storage location 202. Such data paths can be configurable according to configuration data PATH_CFG provided from configuration circuit 206. FIGS. 2A to 2C show various different data path configurations.

FIG. 2A shows an example of a H1→H2 communication path configuration. In such an arrangement, configuration circuit 206 can provide a communication path between first host I/F 208 and second host I/F 210. More particularly, within EP storage locations 202, one EP (EPm) can have an OUT configuration while another EP (EPn) can have an IN configuration. Within switch/control circuit 204, a data path can exist between EPm and EPn and first host I/F 208 and second host I/F 210. Thus, communications from first host I/F 208 to second host I/F 210 can pass through EP storage locations 202, more particularly EPm. Similarly, communications from second host I/F 210 to first host I/F 208 can also pass through EP storage locations 202, more particularly EPn. It is understood that in the example of FIGS. 2A to 2C, while EPm and EPn can be different numbered endpoints, such endpoints could have the same endpoint number, but be fixed in a particular direction.

In this way, a communication path between two hosts can be created that includes endpoint locations.

FIG. 2B shows an example of a H1→F communication path configuration. In such an arrangement, configuration circuit 206 can provide a communication path between first host I/F 208 and function I/F 212. Such an arrangement can allow a host device to access one or more functions available at function I/F 212 essentially directly (e.g., without having to pass through second host I/F 210). More particularly, within EP storage locations 202, one EP (EPm) can have an OUT configuration while another EP (EPn) can have an IN configuration. Within switch/control circuit 204, a data path can exist between EPm and EPn and first host I/F 208 and function I/F 212. Thus, communications from first host I/F 208 to function I/F 212 can pass through EP storage locations 202, more particularly EPm. Similarly, communications from function I/F 212 to first host I/F 208 can also pass through EP storage locations 202, more particularly EPn.

In this way, a communication path between one host and one or more functions can be created that includes endpoint locations.

FIG. 2C shows an example of a H2→F communication path configuration. In such an arrangement, configuration circuit 206 can provide a communication path between second host I/F 210 and function I/F 212. Such an arrangement can allow a host device to access one or more functions available at function I/F 212 essentially directly (e.g., without having to pass through first host I/F 208). More particularly, within EP storage locations 202, one EP (EPm) can exist. Within switch/control circuit 204, a data path can exist between EPm and second host I/F 208 and function I/F 212. Thus, communications between second host I/F 210 and function I/F 212 can pass through EP storage locations 202, more particularly EPm. In one particular arrangement, and as will be described
in other examples herein, in an H2-F configuration, data transfers can include switching the direction of the target endpoint. More particularly, an endpoint can have an IN direction when data is sent from either the second host I/F 210 or function I/F 212, and then can be switched to an OUT configuration to enable data to be received by the function I/F 212 or second host I/F 210.

In this way, a communication path between a second of multiple hosts and one or more functions can be created that includes endpoint locations.

Referring now to FIG. 3, a controller circuit according to another embodiment is shown in a block schematic diagram, and designated by the general reference character 300. In two very particular arrangements, controller circuit 300 can correspond to that shown as 108 in FIG. 1 and/or that shown as 200 in FIGS. 2A to 2C.

In the particular example of FIG. 3, a controller circuit 300 can include some sections like those of FIGS. 2A to 2C, thus like sections are referred to by the same reference character but with the first digit being a “3” instead of a “2.” FIG. 3 can differ from FIGS. 2A to 2C in that EP storage circuit 302 is shown to be logically divided into EP logic memories 302-0 and 302-1. Such a division can represent how some endpoints can be dedicated for particular communication paths. More particularly, EP logic memory 302-0 can include EPs designated for transfer of data between first host I/F 308 and second host I/F 310. EP logic memory 302-1 can include EPs designated for transfer of data between function I/F 312 and either of first host I/F 308 and second host I/F 310.

Switch controller circuit 304 can provide control signals to EP storage circuit 302 to enable switch paths and create various communication paths shown as 314-0 to 314-3 in FIG. 3.

In the example of FIG. 3, a configuration circuit 306 can include a processor circuit that can have a processor and associated instruction memory and input/outputs. Configuration circuit 306 can provide configuration signals to path region 316, which can include the interfaces, memory and logic for enabling the various communication paths between I/Fs (308, 310 and 312). More particularly, a processor equipped configuration circuit 306 can allow configuration data to set up such paths dynamically, via software or firmware for execution by the processor. In one very particular example, a configuration circuit can include an embedded (included in the same substrate as path region 316) 8051 microcontroller, or equivalent. However, alternate embodiments can include different types of microcontrollers or processors to provide the advantageous changing of switch paths based on one or more sets of instructions.

FIG. 3 further differs from FIGS. 2A to 2C in that a first host I/F 308 can particularly include USB physical layer logic, second host I/F 310 and function I/F 312 can both be programmable. Thus, first host I/F 308 can include logic for receiving packet data according to the USB protocol.

Second host I/F 310 can be programmable to accommodate different types of host interfaces. For example, second host I/F 310 can be programmable to enable input signals generated by a second host to be capable of accessing endpoint locations and/or provide control data to other portions of controller circuit 300. Similarly, signals generated by controller circuit 310 can be altered for compatibility with such a second host device.

Function I/F 312 can be programmable to provide access to different types and/or numbers of functions. For example, function I/F 312 can be programmable to enable any of multiple function interfaces. Further I/F 312 can be programmable to provide different versions of a same function (e.g., one function may execute predetermined operations and provide/receive data according to a first bit width, while a different version of the same function can perform the same operation, but provide/receive data according to a larger or smaller bit width).

As noted above, communication paths that can be created with controller circuit are shown as 314-0 to 314-3. Communication path 314-0 can be an H11-H21 communication path passing from first host I/F 308 to second host I/F 310, but completely bypasses function I/F 312. Communication path 314-1 can be an H1-F communication path passing from first host I/F 308 to function I/F 312, bypassing second host I/F 310. Communication path 314-2 can be an H2-F communication path passing from second host I/F 310 to function I/F 312, bypassing first host I/F 308. FIG. 3 also shows a first host-to-control (H1-C) communication path 314-3. Such a communication path 310 can enable a first host to control the controller circuit 300.

Referring still to FIG. 3, a controller circuit 300 is preferably a single integrated circuit formed within a common substrate area 318. Such an arrangement can provide a compact way of interconnecting various different devices.

Referring now to FIG. 4, a controller circuit according to another embodiment is shown in a block schematic diagram, and designated by the general reference character 400. In very particular arrangements, controller circuit 400 can correspond to that shown as 108 in FIG. 1, in FIGS. 2A to 2C, or that shown in FIG. 3.

A controller circuit 400 can include some sections like those of FIGS. 2A to 2C, thus like sections are referred to by the same reference character but with the first digit being a “4” instead of a “2.”

FIG. 4 can differ from FIGS. 2A to 2C in that EP storage circuit 402 includes three set endpoints, as well as a configurable endpoint space. The set endpoints can include an EP0 422-0, configured in an IN and OUT direction, EP1 422-1 configured in an OUT direction, and EP1 422-2 configured in an IN direction. Configurable EP space 422-3 can be configured into one or more endpoints, with varying characteristics. As but a few examples, such characteristics can include buffer size, direction (e.g., IN or OUT), or manner of storing data (e.g., single, double, or quad buffering). The various EPs can provide indications when data blocks of certain sizes have been written into the buffer and/or read from the buffer. Thus, EPs can generate indications when an entire packet of data has been received or read, to thereby indicate the EP is available for more data.

In the particular example of FIG. 4, switch/control circuit 404 can be conceptualized as including a multiplexer (MUX) path 404-0 and a de-MUX path 404-1 that can be controlled according to path configuration data PATH_CFG. Such an arrangement can allow for dynamic switching of data paths between EP storage circuit 402 and either of second host I/F 410 or function I/F 412.

A configuration circuit 406 can include a processor block 406-0 and processor I/F 406-1. A processor block 406-0 can include a processor, associated I/Os, and instruction set. Although not shown in FIG. 4, a configuration circuit 406 can also share a data bus with either second host I/F 410, function I/F 412, or both. Configuration circuit 406 can provide configuration signals for placing controller circuit 400 into various configurations. In the example shown, configuration circuit 406 can provide path configuration data PATH_CFG for controlling switch/control circuit 404 as well as endpoint configuration data EP_CFG for configuring EP space 422-3. Configuration circuit 406 can receive indications from the various EPs that indicate the status of the EPs. Such status
information EP_STATUS can be provided to first host I/F 408. Such an arrangement can allow access to an EP via first host I/F 408 to be temporarily halted in the event the EP is being accessed via the second host I/F 410 or via function I/F 412.

The particular example of FIG. 4 also shows a function section 424 connected to function I/F 412. It is understood that while function section 424 can be embedded into a same substrate as controller circuit 400, preferably such a function section can reside on one or more other integrated circuit devices. Function section 424 can include multiple function circuits, shown as 424-0 to 424-n. Function I/F 412 can allow access to any such functions (424-0 to 424-n) via both first host I/F 408 as well as second host I/F 410.

Having described the structure of a controller circuit in FIG. 4, various operations of such a controller circuit will now be described with respect to FIGS. 5A to 5F.

Referring to FIG. 5A, one example of a H1-H2 communication path is shown for the controller circuit 500 of FIG. 4. In the particular arrangement shown, EP space 422-3 has been configured to include an EP2, configured in the IN direction, and an EP4 configured in an OUT direction. Further, switch/control path 404 has been configured to provide a communication path from second host I/F 410 to EP storage circuit 402, and from EP storage circuit 402 to second host I/F 410. In such a configuration, an OUT direction for H1-H2 can be formed, shown by solid line 414-0 (OUT). OUT path 414-0 can be formed from first host I/F 408, to EP4, through de-MUX path 404-1, to second host I/F 410. In addition, an IN direction for H1-H2 can be formed, shown by dashed line 414-1 (IN). IN path 414-0 can be formed from second host I/F 410, through MUX path 404-0, to EP2, and to first host I/F 410.

In this way, a controller circuit like that of FIG. 4 can provide a H1-H2 path.

Another feature of the arrangement like of FIG. 4 can be the use of a single physical endpoint to access multiple logical endpoints. One example of such an arrangement is shown in FIGS. 53-0 and 53-1. In the particular arrangement shown, EP space 422-3 has been configured to include an EP2 configured in the IN direction, and an EP4 configured in an OUT direction. Switch/control path 404 can be used to direct a data path from a same physical endpoint (e.g., EP4) to different locations, where such different locations form different logical endpoints. In the example shown a first logical endpoint LEP4a can be a second host I/F 410, and a second logical endpoint LEP4b can be function circuit 424-0.

FIGS. 53-0 shows OUT communication paths that can be created for logical endpoints LEP4a and LEP4b. Both OUT communication paths include a first portion 414-0/1 from first host I/F 408 to EP4. De-MUX path 404-1 can be controlled to provide access to different logical endpoints. Thus, a path 414-0 to logical endpoint LEP4a (shown by a dashed line) can include EP4 and de-MUX 404-1 having a first configuration. A path 414-1 to logical endpoint LEP4b (shown by a dashed and dotted line) can include EP4 and de-MUX 404-1 having a second configuration.

FIG. 5B-1 shows IN communication paths that can be created for logical endpoints LEP4a and LEP4b. MUX path 404-0 can be controlled to provide paths from different logical endpoints. Thus, a path 414-0' from logical endpoint LEP4a can include MUX 404-0 having a first configuration and EP2. A path 414-1' to from logical endpoint LEP4b can include MUX 404-0 having a second configuration and EP2. Both IN communication paths include a path 414-0/1' from EP2 to host I/F 408.

In this way, a controller circuit can provide multiple logical endpoints for a first host I/F utilizing a single physical endpoint.

Another example showing the mapping of multiple logical endpoints to a same physical endpoint is shown in FIGS. 5C-0 and 5C-1. In the particular arrangement shown, EP space 422-3 has been configured to include an EP2. A direction of EP2 can be switched to provide different logical endpoints from this one physical endpoint. In the example shown a first logical endpoint LEP2a can be a first host I/F 408, and a second logical endpoint LEP2b can be function circuit 424-0.

FIGS. 5C-0 shows an IN communication path for logical endpoints LEP2a. EP2 can be configured in an IN direction, and MUX path 404-0 can connect second host I/F 410 to EP storage circuit 422-3. A path 414-0 to logical endpoint LEP2a can include MUX 404-0 and EP2 (configured in the IN direction).

FIG. 5C-1 shows an IN communication path with respect to logical endpoint LEP2b. EP2 can be initially be configured in an IN direction, and MUX path 404-3 can connect second host I/F 410 to EP storage circuit 422-3. In such an arrangement data can be written from second host I/F 410 to EP2 through MUX 404-0. Subsequently, a direction of EP2 can be switched from IN to OUT. A communication path can then be created from EP2, through function I/F 412, and on to function circuit 424-0. At the same time, a configuration circuit 406 can provide EP status information that indicates EP2 should not be accessible by first host I/F 408 (shown as “EP2 NAK”). Thus, a path 414-2 to logical endpoint LEP2b can include MUX 404-0, EP2, de-MUX 404-1 and function I/F 412.

While FIGS. 5C-0 and 5C-1 show paths to different logical endpoints from second host I/F 410, the same approach can be used to provide paths from such logical endpoints to second host I/F 410.

In this way, a controller circuit can provide multiple logical endpoints for a second host I/F utilizing a single physical endpoint.

An arrangement like of FIG. 4 can also allow a single host device to access multiple functions. Examples of arrangements are shown in FIGS. 53-0 and 5E.

FIG. 5D shows OUT communication paths between different function circuits 424-0 and 424-1 and first host I/F 408. In the example shown, EP4 can be configured in an OUT direction. De-MUX 404-1 can provide a communication path from EP storage circuit 422-3 to function I/F 412. A path 414-1' to a first function circuit 424-0 can include EP4 (OUT) 422-1, de-MUX 404-0, and function I/F 412. In a similar fashion, an OUT communication path 414-1' to second function circuit 424-n can include EP2, de-MUX 404-0, and function I/F 412.

FIG. 5E shows communication paths between different function circuits 424-0 and 424-1 and second host I/F 410. To access function circuit 424-0, initially EP2 can be configured in an IN direction, and MUX 404-0 can provide a communication path from second host I/F 410 and EP storage circuit 422-3. Subsequently, the direction of EP2 can be changed, and de-MUX 404-1 can provide a communication path from EP storage circuit 422-3 and function I/F 412. To access function circuit 424-n, operations can be essentially the same as that for function circuit 424-0, but include the direction switching of EP4. A communication path to function circuit 424-0 is shown as 414-2F0. A communication path to function circuit 424-n is shown as 414-2Fn.

While FIGS. 5D and 5E show paths having a particular direction, the same communication link in the other direction (to a host) is understood from the other examples above.
In this way, a controller circuit can allow any of multiple host devices (via host I/Fs) to access any of multiple functions.

An arrangement like of FIG. 4 can also allow one host device to provide control information to the controller, while, at the same time, a second host accesses functions independently of the first host. FIG. 5F shows one very particular example of such an operation.

As shown in FIG. 5F, a first host-to-control (H1-C) communication path 414-3 can utilize EP0 to provide control information to configuration circuit 406. At the same time, a second host-to-function (H2-F) communication path 414-1 can exists.

In this way, a controller circuit can establish a control communication path to one host while allowing another host to access available functions.

As noted above, embodiments of the present invention can include a configurable function interface. A first particular example of a configurable function interface is shown in FIG. 6, and designated by the general reference character 600.

A configurable function I/F 600 can include a number of physical input/outputs (I/Os) 602, an I/O switch circuit 604, a first function I/F circuit 606, and a second function I/F circuit 608. In the particular example of FIG. 6, physical I/Os 602 can include first control inputs and outputs 602-0, configurable I/Os 602-1, second control inputs and outputs 602-2, and I/Os 602-3. Preferably, physical I/Os can include bond pads of a same integrated circuit device that contains a controller according embodiments shown herein, and equivalents.

An I/O switch circuit 604 can selectively connect configurable I/Os 602-1 to either first function I/F circuit 606 or second function I/F circuit 606 according to configuration information FUNC_CFG.

First function I/F circuit 606 can translate a first type of external signal set into signals compatible with circuits internal to a controller circuit. Similarly, second function I/F circuit 608 can translate a second type of external signal set into signals compatible with circuits internal to controller circuit. As but one example, function I/F circuits (606 and 608) can enable data transfers from endpoint locations to predetermined addresses of devices attached to function I/F 600. The very particular example of FIG. 6 shows a second I/F circuit 606 that can be configured between two different modes of operation. A low I/O count mode 608-0 and a high I/O count mode 608-1. In a high I/O count mode 608-1, second I/F circuit 606 can operate according to both I/Os sets 602-1 and 602-3. In a low I/O count mode 608-0, second I/F circuit 606 can operate according to I/Os set 602-3.

In one very particular example, a first I/F circuit 606 can be a storage port compatible with the SD Memory Card Specification (SD), promulgated by the SD Card Association, the MultiMediaCard System Specification (MMC), promulgated by the MMC Association, and the CE-ATA Standard, promulgated by the CE-ATA Working Group. A second I/F circuit 608 can be a storage port compatible with a NAND memory device, such as those complying with the Open NAND Flash Interface Working Group. More particularly, second I/F circuit 608 can be configurable to operate to operate with one number of I/Os (e.g., x8) or another number of I/Os (e.g., x16).

In this way, a configurable function I/F can accommodate one or more different mass storage media devices of differing types.

A second particular example of a configurable function interface is shown in FIG. 7, and designated by the general reference character 700. The configurable I/F 700 of FIG. 7 can include general purpose I/Os 702 and function I/F circuits 704-0 and 704-1. General purpose I/Os 702 can be configured according to function configuration data FUNC_CFG, to provide inputs and/or outputs to function I/F circuits (704-0 to 704-1). In the particular example shown, function I/F circuits can include a storage I/F 704-0 and a modem I/F 704-1. A storage I/F 704-0 can provide in interface compatible for accessing a memory device. A modem I/F 704-1 can include an interface compatible for transferring data communicated via a modem.

Of course, while FIGS. 6 and 7 show interfaces to particular functions, the present invention should not necessarily be construed as being limited to any particular function interface.

Just as function interfaces can be programmable, in the embodiments described herein, host interfaces can be programmable. An example of such an arrangement is shown in FIG. 8.

A programmable host I/F 800 can include a number of host physical input/outputs (I/Os) 802, a clock circuit 804, data input latches 806, address input latches 808, control input latches 810, data output buffers 806, control output buffers 810, and I/O switch circuit 812, a control input map circuit 814, a control output map circuit 816, and a programmable state machine 818.

In the particular example of FIG. 8, host physical I/Os 802 can include a clock input 802-0, data I/Os 802-1, address inputs 802-2, control inputs 802-3, control outputs 802-4, and general purpose I/Os 802-5. Preferably, host physical I/Os can include bond pads of a same integrated circuit device that contains a controller according embodiments shown herein, and equivalents.

A clock circuit 804 can receive a clock signalCLK_IN, and generate an internal clock signal CLK. In particular arrangements, a clock circuit 804 can be a buffer circuit, or alternatively, can include frequency locking circuits, such as a phase lock loop (PLL) or delay lock loop (DLL).

Data input latches 806-0 can latch data input signals according to clock signal CLK. Similarly, address input latches 808 can latch address values, and control input latches 810-0 can latch control input values according to clock signal CLK.

Data output buffers 806-1 can drive data values on data I/Os 802-1, also in synchronism with clock signal CLK. Control output buffers 810-1 can drive control data on control outputs 802-4 according to clock signal CLK.

I/O switch circuit 812 can selectively enable signal paths from GPI/Os 802-5 to programmable state machine 818, according to host configuration data HOST_CFG.

Control input map circuit 814 can map incoming control signals to particular internal nodes of a controller circuit and/or logically combine such signals to generate internal control signals. In the reverse manner, control output map circuit 816 can map signals at internal nodes of a controller circuit to output control signals and/or logically combine such internal signals to generate output control signals. Such a mapping can be based upon host configuration data HOST_CFG.

A programmable state machine 818 can be programmed to generate a set of output signals in response to predetermined input signals, and set of input signals in response to predetermined output signals. In one particular case, such an arrangement can include "waveform descriptors" which can include both unconditional and conditional steps that are executed according to predetermined signal sets.

In this way, a programmable interface can allow a controller circuit to accommodate various types of host devices.
A controller circuit according the various embodiments can base timing on a generated clock signal. To accommodate various system speeds, it may be desirable to include a clock circuit that can generate or accommodate clock signals of varying frequencies. One particular example of such an arrangement is shown in FIG. 9.

Referring to FIG. 9, a clock circuit is shown in block schematic diagram and designated by the general reference character 900. A clock circuit 900 can include a crystal oscillator circuit 902, a level translator circuit 904, an external clock buffer 906, and a clock signal selector 908. Optionally, a clock circuit 900 can include a phase lock loop 910.

A crystal oscillator circuit 902 can be connected to a crystal and other passive components by way of a crystal input terminal 912-0 and a crystal output terminal 912-1, which are preferably physical inputs to an integrated circuit (e.g., pins).

Crystal oscillator circuit 902 can generate a periodic output signal fxout having a frequency dependent upon an attached crystal. Different oscillating frequencies can be accommodated by selection of different components based on signal XCLKSEL. A crystal oscillator circuit 902 can receive a power supply voltage VDDX, which can be greater than that of other components of a controller circuit.

A level translator circuit 904 can translate and buffer output signal fxout to levels suitable for the other portions of the clock circuit 900.

An external clock buffer 906 can buffer a clock signal received at an external clock input terminal 914. Preferably, such a clock input terminal is a physical input to an integrated circuit.

A clock signal selector 908 can select a clock signal provided from the external clock buffer 906 or crystal oscillator circuit 902 as an input clock signal CLK_IN based on a clock configuration signal CLK_CFG. Optionally, input clock signal CLK_IN can be provided as an input to a phase lock loop circuit 910, which can multiply and/or divide signal CLK_IN to generate a device clock signal CLK_IN. In the particular arrangement shown, clock division multiplication can occur according to values CFG_CLK.

In this way, a controller circuit can include a clock signal that can operate according to clock signals having a selectable frequency. Further such a clock signal can be generated by an onboard oscillator, or received via a clock input.

As shown in various embodiments above, a controller circuit can provide various communication paths by controlling a data path between an endpoint storage circuit and either a second host I/F or a function I/F. Additional communication paths can be formed by dynamically switching directions of endpoints. As previously noted, it may be advantageous to provide changes in the configuration of such paths in a dynamic fashion. One example of a configuration circuit for providing such dynamic control is shown in FIG. 10, and designated by the general reference character 1000.

A configuration circuit 1000 can include a processor 1002, an instruction memory 1004, control registers 1006, and an address/data bus 1008. In very particular examples, a configuration circuit 1000 can correspond to any of configuration circuits shown as 206, 306 or 406 in other embodiments above.

A processor 1002 can execute instructions stored in an instruction memory 1004 and can provide input/output signals on configuration I/O lines 1010. In the very particular example of FIG. 10, a processor 1002 can include a microcontroller. An instruction memory 1004 can store instructions for execution by processor 1002.

It is noted that instruction memory 1004 can be connected to address/data bus 1008, and address bus 1008 can be connected to a second host I/F, EPs, and/or a function I/F. Such an arrangement can allow instructions to be stored in instruction memory 1004 by either a first host (e.g., via an EP), a second host, or a function. Instructions stored within instruction memory 1004 can be firmware or software. This configuration can allow such firmware or software to be loaded via a first host I/F, second host I/F, or even a function I/F.

In the particular example of FIG. 10, configuration of EPs and/or communication paths (e.g., MUXing and de-MUXing from EPs) can be established by values stored in control registers 1006. Thus, a processor 1002 can write values to control registers to thereby configure EPs and/or data paths. Optionally, such control registers can be accessible by way address/data bus 1008. Such an arrangement can enable a controller circuit to be configured by either host I/F, or by way of function I/F.

Alternatively, a processor 1002 can provide configuration data by way of I/O lines, rather than control registers.

In this way, a control circuit can configure a controller circuit according to sequences contained within firmware or software.

In controller circuit configurations like that of FIG. 4, the formation of direct communication paths between host devices and common functions can include sharing endpoints. In such an arrangement, it may be desirable to ensure that one host is prevented from accessing an endpoint while another host or function is using the endpoint. Circuits for such an arrangement are shown in FIG. 11.

FIG. 11 shows a host input circuit according to one embodiment. Host input circuit 1100 can include a physical layer logic circuit 1102, an endpoint detect circuit 1104, and an acknowledgement circuit 1106. A physical layer logic circuit 1102 can receive data from a first host, and place such data on a DATA_OUT bus 1106. Similarly, data from a controller circuit can be received on a DATA_IN bus 1108. In one very particular example, a host input circuit 1100 can be a USB physical layer circuit that can receive an incoming serial data stream in packets, and format such data for storage at endpoint locations. Such a circuit can also format outgoing data into packets, and output such data in a serial data stream.

An endpoint detect circuit 1104 can examine outgoing data from a host for a destination endpoint. Such an endpoint value can be provided to acknowledgement circuit 1106. Acknowledgement circuit 1106 can receive endpoint status information EP_STATUS and a destination endpoint value from endpoint detect circuit 1104. If acknowledgement circuit 1106 determines that a destination endpoint matches an endpoint currently in use by another host or function, it can output a “no acknowledge” indication (NAK) to physical layer logic circuit 1102. In response, physical layer logic circuit 1100 can issue a no acknowledgement back to the host.

In this way, a controller circuit can handle accesses from one host while an endpoint is in use by another host or function.

Embodiments of the present invention can be well suited to performing various other steps or variations of the steps recited herein, and in a sequence other than that depicted and/or described herein.

For purposes of clarity, many of the details of the various embodiments and the methods of designing and manufacturing the same that are widely known and are not relevant to the present invention have been omitted from the following description.

It should be appreciated that reference throughout this specification to “one embodiment” or “an embodiment” means that a particular feature, structure or characteristic described in connection with the embodiment is included in at
least one embodiment of the present invention. Therefore, it is emphasized and should be appreciated that two or more references to "an embodiment" or "one embodiment" or "an alternative embodiment" in various portions of this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures or characteristics may be combined as suitable in one or more embodiments of the invention.

Similarly, it should be appreciated that in the foregoing description of exemplary embodiments of the invention, various features of the invention are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure aiding in the understanding of one or more of the various inventive aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are explicitly claimed. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment. Thus, the claims following the detailed description are hereby expressly incorporated into this detailed description, with each claim standing on its own as a separate embodiment of this invention.

It is also understood that the embodiments of the invention may be practiced in the absence of an element and/or step not specifically disclosed. That is, an inventive feature of the invention can be eliminated of an element.

Accordingly, while the various aspects of the particular embodiments set forth herein have been described in detail, the present invention could be subject to various changes, substitutions, and alterations without departing from the spirit and scope of the invention.

What is claimed is:

1. A controller circuit comprising:
an endpoint buffer circuit having a plurality of storage locations configurable as endpoints according to a USB protocol; and
a data switching circuit coupled to the endpoint buffer circuit configurable to provide communication paths that enable a first host interface (I/F) and a second host I/F to access at least an individual function I/F of one or more function I/Fs of the controller circuit configured to allow access to at least one predetermined circuit function, and to enable first and second host I/Fs's to communicate with one another, wherein the communication paths include a first communication path that enables the first host I/F to access at least the individual function I/F without passing through the second host I/F and a second communication path that enables the second host I/F to access at least the individual function I/F without passing through the first host I/F, and wherein the data switching circuit comprises a multiplexer (MUX) and a de-multiplexer (DE-MUX) coupled to the endpoint buffer circuit, the second host I/F and the at least one individual function I/F to selectively provide the communication paths.

2. The controller circuit of claim 1, wherein:
the individual function I/F is coupled to any selected from the group consisting of a storage device or input/output device.

3. The controller circuit of claim 1, wherein:
the data switching circuit can include any selected from the group consisting of a shared data bus or the multiplexer selectively accessible by at least the second host I/F and the individual function I/F.

4. The controller circuit of claim 1, wherein:
the controller circuit is configurable to provide a first host-to-second host transmission path that includes portions of the data switching circuit and portions of the endpoint buffer circuit.

5. The controller circuit of claim 1, wherein:
the controller circuit is configurable to provide a third communication path between the individual function I/F and both the first host I/F and the second host I/F that includes portions of the data switching circuit and portions of the endpoint buffer circuit.

6. The controller circuit of claim 1, wherein:
the at least one function I/F comprises a plurality of function I/Fs that are each configurable to provide one or more signals for accessing at least one different predetermined function, the controller circuit being configurable to provide an individual first host I/F communication path between the first host I/F and at least two of the plurality of function I/Fs.

7. The controller circuit of claim 1, wherein:
the at least one function I/F comprises a plurality of function I/Fs that are each configurable to provide one or more signals for accessing at least one different predetermined function, the controller being configurable to provide an individual second host I/F communication path between the second host I/F and at least two of the plurality of function I/Fs.

8. The controller circuit of claim 1, wherein:
the endpoint buffer circuit is configurable to have at least one physical endpoint corresponding to an endpoint identification value of the USB protocol; and
the data switching circuit is configurable to provide at least two different communication paths from the one physical endpoint as different logical endpoints corresponding to the one physical endpoint.

9. The controller circuit of claim 1, wherein:
the individual function I/F is programmable between at least two different function I/Fs, each different function I/F enabling access to at least one other predetermined circuit function.

10. The controller circuit of claim 9, wherein:
the at least two different functions include accesses to different types of solid state storage media.

11. The controller circuit of claim 1, wherein:
the second host I/F is programmable between at least two different configured host I/Fs, each different configured host I/F enabling communication with a different type of host.

12. The controller circuit of claim 1, further including:
a processor coupled to a memory and configurable to execute instructions stored in the memory and outputs configuration data; and
wherein the data switching circuit is configurable according to the configuration data.

13. The controller circuit of claim 1, wherein the first host I/F comprises:
an endpoint detector configurable to determine an endpoint value of a communication received at the first host I/F, and
a response circuit coupled to the endpoint detector and configurable to receive endpoint status information, the response circuit providing a predetermined indication when the endpoint value corresponds to an endpoint indicated as in use by at least one function I/F of the one or more function I/Fs or the second host I/F by the endpoint status information.
14. The controller circuit of claim 1, wherein:
the data switching circuit is configurable to provide a control communication path between the first host I/F and a control endpoint, and at the same time, the second communication path.

15. The controller circuit of claim 1, further including:
a processor circuit configurable to execute instructions stored in a processor memory and outputs configuration data, the processor memory being coupled to any selected from the group of: the first host I/F, the second host I/F, or at least one of the one or more function I/Fs; and

wherein the data switching circuit is configurable according to the configuration data.

16. The controller circuit of claim 1, further including:
a timing circuit for generating a timing signal for the controller circuit, the timing circuit comprising at least one crystal input operable for receiving a crystal; and

an oscillator circuit for generating an oscillating signal according to a crystal at the at least one crystal input, the oscillator circuit configured to accommodate any of a plurality of crystals of different resonance frequencies.

17. A method comprising the steps of:
in one integrated circuit,

providing at least a first data communication path between a first host interface (I/F) and at least a first endpoint storage location;

providing a second data communication path between the first host I/F and a second endpoint storage location;

configuring the second communication data path between the second endpoint storage location and a controller; and

configuring a third data communication path between at least the first endpoint storage location and a second host I/F and between at least the first endpoint storage location and a function I/F, wherein the second host I/F is programmable to accommodate different interface types, and wherein the function I/F is programmable to provide access to one or both of different types of functions or different versions of an individual function, and wherein the second data communication path and the third data communication path can be utilized simultaneously.

18. The method of claim 17, further including:
in a communication between the first host I/F and the second host I/F or a transfer between the first host I/F and the function I/F, storing data in at least one endpoint storage location configured for the direction of communication.

19. The method of claim 18, wherein:
in a communication between the second host I/F and the function I/F, storing data in at least one endpoint storage location configured for in a first direction, and switching the direction of the endpoint storage location.

20. The method of claim 17, wherein:
configuring the data communication path includes creating predetermined signal paths according to configuration information output from a processor that executes instructions in an instruction memory; and

in a configuration operation, loading instruction data into the instruction memory from one or more of: the first host I/F, the second host I/F, or the at least one function I/F.

21. A controller circuit device, comprising:
an integrated circuit substrate that includes

a first host interface (I/F) operable to communicate with a first type host device, a second host I/F operable to communicate with a second type host device, and at least one function I/F operable to communicate with at least one predetermined function; and

a control and memory circuit configurable to provide a plurality of communication paths, including at least a first host I/F to second host I/F path that includes a portion of an endpoint buffer memory but does not include the at least one function I/F, a first host I/F to function I/F path that includes a portion of the endpoint buffer memory but does not include the second host I/F, and a second host I/F to function I/F path that includes a portion of the endpoint buffer memory but does not include the first host I/F, and further comprising a multiplexer (MUX) and a de-multiplexer (DE-MUX) coupled to the endpoint buffer memory, the second host I/F and the at least one function I/F to selectively provide at least some of the plurality of communication paths.

22. The controller circuit device of claim 21, wherein:
the first host I/F is a universal serial bus (USB) I/F.

* * * * *